

Symplectic, linearly-implicit and stable integrators, with applications to constrained dynamics

Molei Tao, Houman Owhadi, Jerrold E. Marsden

January 16, 2013

Abstract

We introduce a family of symplectic, linearly-implicit and stable integrators for mechanical systems. When used in conjunction with penalty methods (i.e., methods that consist in replacing constraints by stiff potentials), these integrators accelerate the numerical simulation of mechanical systems with holonomic constraints by employing coarse timesteps and bypassing the resolution of nonlinear systems. Although penalty methods are well known and widely employed, a general and rigorous proof of their accuracy appeared to be lacking; such a proof is also provided in this paper.

1 Introduction and main results

In this paper, we introduce a new family of symplectic, linearly-implicit and stable integrators for the following mechanical dynamics

$$\begin{cases} \dot{x} = v \\ M\dot{v} = -\nabla V(x) \end{cases} \quad (1)$$

where $V \in \mathcal{C}^2(\mathbb{R}^n)$ is a potential energy function and M a $n \times n$ constant, symmetric mass matrix. This new family of integrators (SyLiPN) is parameterized by a constant β and is of the form

Integrator 1. *Symplectic Linearized Push-forward Newmark (SyLiPN):*

$$\begin{cases} x_{k+1} &= x_k + hv_k + \frac{1}{2}h^2a_k \\ v_{k+1} &= v_k + \frac{1}{2}h(a_k + a_{k+1}) \\ a_k &= -M^{-1}\nabla V(x_k) - M^{-1}\text{Hess}V(x_k)\beta h^2a_k \end{cases} \quad (2)$$

Remark 1.1. Notice that the third line, i.e., the force evaluation, can be rewritten as

$$a_k = -(M + \text{Hess}V(x_k)\beta h^2)^{-1}\nabla V(x_k) \quad (3)$$

This evaluation, however, should be executed by solving a symmetric linear system instead of inverting a matrix for computational efficiency. In this sense, SyLiPN is linearly implicit.

Remark 1.2. We refer to Integrator 4 for a simplification of SyLiPN when V can be decomposed as the sum of a soft and a stiff potential.

In Section 2, we will show how SyLiPN is obtained by linearizing the Push-forward of the Newmark family of implicit integrators [24, 30, 22]. We will also prove (in Section 2) that SyLiPN has the following properties.

Theorem 1.1. *SyLiPN (Integrator 1):*

1. *Is linearly unconditionally stable if $\beta \geq 1/4$.*
2. *Has a 3rd order local error.*
3. *Is symplectic.*

Our motivation for developing SyLiPN originates from (but is not limited to) a need for such integrators in penalty methods, i.e., methods based on replacing holonomic constraints with stiff potentials [33, 36, 26] as described below (we note that SyLiPN is also relevant to applications, such as molecular dynamics, where stiff potentials are replaced with holonomic constraints [7, 28]).

Let

$$\mathcal{S}(q(t)) := \int_a^b \frac{1}{2} \dot{q}(t)^T M \dot{q}(t) - V(q(t)) dt \quad (4)$$

be the action functional associated with the mechanical dynamical system (1). The evolution of this mechanical system under the holonomic constraint $g(q) = 0$ can be obtained as the critical trajectory on the constrained manifold, i.e., the solution of:

$$\delta \mathcal{S} / \delta q = 0 \text{ and for all } t, q(t) \in g^{-1}(0) \quad (5)$$

This constrained dynamic can also be shown as the solution to the following Differential Algebraic Equation system:

$$\begin{cases} M \ddot{q} = -\nabla V(q) + \lambda^T \nabla g(q) \\ g(q) = 0 \end{cases} \quad (6)$$

The idea behind penalty methods is to replace the rigid constraints by stiff potentials that enforce the holonomic constraints up to small deviations. More precisely, the potential energy $V(q)$ is modified to be $V(q) + \frac{1}{2} \omega^2 g(q)^T g(q)$, and the solution of the constrained system (5) is approximated by the solution of the following unconstrained mechanical system.

$$M \ddot{q} = -\nabla V(q) - \omega^2 g(q)^T \nabla g(q) \quad (7)$$

where ω is chosen to be large enough for accuracy. Paraphrasing [26], the problem with this approach is that “*as a result of stiffness, the numerical differential equation solver*

takes very small time steps, using a large amount of computing time without getting much done”. With SyLiPN, this issue can now be solved by employing coarse timesteps and bypassing the resolution of nonlinear systems. Up to the authors’ knowledge, SyLiPN is the first symplectic, stable integrator for such systems that does not require the resolution of a nonlinear system at each timestep.

Although penalty methods are well known and widely employed, a rigorous justification of this accuracy appeared to be lacking. Our analysis shows that the solution of (7) converges to the solution of (6) as $\omega \rightarrow \infty$ in the sense of two-scale flow convergence (see Definition 1.1 in [31]). More precisely, we will prove the following theorem in Section 3 (throughout this analysis, ‘bounded’ means having a norm no larger than a constant independent of ω).

Theorem 1.2. *Denote by $q^\omega(t)$ the solution to (7) with $q^\omega(0) = q_0$ and $\dot{q}^\omega(0) = \dot{q}_0$ (where $g(q_0) = 0$ and $\frac{d}{dt}g(q_0) = \nabla g(q_0) \cdot \dot{q}_0 = 0$). Suppose that M is non-singular, $V(\cdot)$ is bounded from below, $V(q)$ diverges towards infinity as $|q| \rightarrow \infty$, $V(\cdot)$ and $g(\cdot)$ are C^2 with bounded derivatives, and for all $q \in g^{-1}(0)$, $\nabla g(q)$ has a constant rank equal to the codimension of the constrained manifold, then*

$$\lambda(t) := - \lim_{T \rightarrow 0} \lim_{\omega \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \omega^2 g(q^\omega(s)) ds \quad (8)$$

exists. Also, the solution to

$$\begin{cases} M\ddot{q}(t) = -\nabla V(q(t)) + \lambda(t)^T \nabla g(q(t)) \\ g(q(t)) = 0 \\ q(0) = q_0 \\ \dot{q}(0) = \dot{q}_0 \end{cases}, \quad (9)$$

which we denote by $q(t)$ ($q(t)$ is also the solution to (6)), exists and satisfies

$$q^\omega \xrightarrow{F} q \quad (10)$$

in the sense that for all bounded $t \geq 0$ and all bounded and uniformly Lipschitz continuous test function φ ,

$$\lim_{T \rightarrow 0} \lim_{\omega \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \varphi(q^\omega(s)) - \varphi(q(s)) ds = 0 \quad (11)$$

In section 4, we will investigate three numerical examples: (1) double pendulum (2) a chain of many pendulums, i.e., an approximation of continuous ropes, hairs and soft tree branches (these examples share the same constraints, differ in the soft potentials, and are of significant importance in computer graphics) (3) the dynamics of water cluster, i.e., multiple water molecules that interact with each other. Numerical results show the accuracy of SyLiPN and significantly improved computation time for all three examples.

1.1 On implicit methods, penalty methods and constrained dynamics

Implicit methods are widely used for stable integrations, and the symplectic members of them have been successful in long time integrations of mechanical systems [9]. Implicit methods require solving nonlinear systems, which oftentimes cost more than one iteration in a nonlinear solver (e.g., Newton’s method [25]) when symplecticity is desired. Indeed, for symplectic integrators, if one solves the nonlinear updating equations partially (for instance, by carrying out only the first step in a Newton’s solver) or linearizes the nonlinearity (which can be shown to be equivalent to the former), in most cases symplecticity is lost. Such a loss results in unsatisfactory long time performances, including drifts in energy and momentum (imagine a tennis video game in which the ball does not bounce back from the racquet constrained in the player’s grip), or even instability. SyLiPN is linearly-implicit in the sense that it only requires the resolution of a linear system at each integration step, which can be computed efficiently by methods provided in, for instance, [2] and references therein. SyLiPN inherits both the stability and the symplecticity from Newmark, however at a much lower computational cost. In other words, the choice of Push-forward Newmark in SyLiPN allows the nonlinear implicit updates to be solved by the first Newton iteration without losing symplecticity.

As an important application of SyLiPN, mechanical systems with holonomic constraints can be simulated more efficiently than by existing methods, which include generalized coordinate approaches (e.g., [13]) and Lagrange multiplier methods (e.g., [27, 1, 10, 23, 17]) — both are implicit in general cases. For doing so, we first approximate the constrained dynamics (5) by the modified system (7) in which rigid algebraic constraints are replaced by stiff nonlinear springs, and then employ SyLiPN to integrate the modified system (7) with a large timestep (which does not require the resolution of the stiffness in the equations). The strategy that consists in replacing holonomic constraints by stiff potentials (and vice versa) is, by now, widely acknowledged and used. In molecular dynamics, for instance, stiff oscillatory molecular bonds are replaced with rigid bond length constraints for the purpose of allowing large timesteps (e.g., [7, 28]). The reverse point of view, which corresponds to replacing rigid bond length constraints with stiff oscillatory bonds, is of significant and practical importance in computer graphics and related fields (we refer for instance to [33, 36, 26]).

Traditional methods for simulating the constrained mechanical system defined by (5) include (but are not limited to): introducing generalized coordinates on the constrained manifold (e.g., [13]), or using Lagrange multipliers (e.g., SHAKE [27], RATTLE [1], SETTLE [10], LINC [23], M-SHAKE [17]). The equivalence between these two approaches is now well-established (see for instance [35] for a discussion of the mathematical justification). These numerical methods (e.g., [13, 27, 1, 10, 23, 17]) allow a $o(1)$ integration step, but they also require solving nonlinear systems at each step (which significantly slows down computations). Another strategy would be to use a classical symplectic integrator (such as Velocity Verlet) to explicitly simulate the modified mechanical system (7) at the cost of using $o(\omega^{-1})$ timesteps. The best strategy depends on the system: if the constraints are high dimensional, an explicit integration, even with small steps, will still be faster, because the nonlinear solve is too expensive; however, if there are just a

few constraints, generalized coordinate and Lagrange multiplier methods will be more efficient.

SyLiPN (Integrator 1) allows for large ($o(1)$) integration timesteps with the limited cost of a linear solve per iteration (instead of a nonlinear one). SyLiPN also remains accurate with $o(1)$ timestep when applied to (7) because the stiffness in this system results in a fast dynamics that (locally) converges to a point distribution (see analysis in Section 3), whose effective contribution to the slow dynamics can therefore be captured by an implicit method [21]. Therefore, SyLiPN will be both efficient and accurate for constrained dynamics. Let us also observe that, although efficient, M-SHAKE [17] is limited to systems with distance constraints, and SyLiPN does not suffer from this restriction: it is a more general approach that can be applied to arbitrary holonomic constraints.

2 SyLiPN: a family of symplectic, linearly-implicit and stable integrators

The following Newmark family of algorithms have been widely used in structure dynamics [24]:

Integrator 2. *Newmark:*

$$\begin{cases} q_{k+1} &= q_k + h\dot{q}_k + \frac{h^2}{2}[(1-2\beta)a_k + 2\beta a_{k+1}] \\ \dot{q}_{k+1} &= \dot{q}_k + h[(1-\gamma)a_k + \gamma a_{k+1}] \\ a_k &= M^{-1}(-\nabla V(q_k)) \end{cases} \quad (12)$$

It was known [22] that Newmark is 2nd-order accurate when $\gamma = 1/2$ and 1st-order otherwise, and it is generally implicit when $\beta \neq 0$. It was also shown [15] that Newmark is variational for arbitrary β when $\gamma = 1/2$ (we will restrict ourselves to this case throughout this paper). However, it is worth noticing that the symplectic form that Integrator 2 preserves is not the canonical one. In fact, it was shown [30, 22] that if one pushes forward the Newmark integrator by a coordinate transformation $\eta : TQ \rightarrow TQ$ defined as

$$(x, v) := \eta(q, \dot{q}) = (q + \beta h^2 M^{-1} \nabla V(q), \dot{q}) \quad , \quad (13)$$

then we obtain an integrator that preserves the canonical symplectic form on T^*Q :

Integrator 3. *Push-forward Newmark:*

$$\begin{cases} x_{k+1} &= x_k + hv_k + \frac{1}{2}h^2 a_k \\ v_{k+1} &= v_k + \frac{1}{2}h(a_k + a_{k+1}) \\ a_k &= -M^{-1} \nabla V(x_k + \beta h^2 a_k) \end{cases} \quad (14)$$

The Newmark and Push-forward Newmark schemes can be shown to be unconditionally linearly stable if $\beta \geq 1/4$ [5, 30], and Newmark was further shown to be nonlinearly stable in the same case under several assumptions [12]. In addition, Push-forward Newmark was shown to be stable near stable fixed points in general nonlinear settings, unless specific resonances occur [29]. Nevertheless, there are nonlinear cases in which Newmark is no longer stable [6, 19]. In fact, few convergent methods are unconditionally stable for arbitrary nonlinear systems to the authors' knowledge (e.g., see a discussion in [37]).

Now, linearize the nonlinear force in Push-forward Newmark at each step by Taylor expansion at x_k , so that a nonlinear implicit equation becomes a linear one. One obtains Integrator 1, i.e., SyLiPN.

Remark 2.1. Since Newmark or Push-forward Newmark requires solving a nonlinear system at each step, SyLiPN exhibits a speed advantage. We will quantify this advantage numerically in Section 4.

Next, we will show that SyLiPN is stable, convergent, and symplectic. SyLiPN is, so far, the only method with these properties that does not require the resolution of a nonlinear system at each coarse timestep.

Theorem 2.1 (Stability). *SyLiPN (Integrator 1) is linearly unconditionally stable if $\beta \geq 1/4$.*

Proof. The definition of linear unconditional stability for mechanical system is that when $V(x) = \frac{1}{2}x^TKx$ with K positive definite, the integrator produces bounded trajectories for all time for all integration step h .

Plotting V into the SyLiPN updating rule (2), we obtain

$$\begin{cases} x_{k+1} = x_k + hv_k + \frac{1}{2}h^2a_k \\ v_{k+1} = v_k + \frac{1}{2}h(a_k + a_{k+1}) \\ Ma_k = -Kx_k - K\beta h^2a_k \end{cases} \quad (15)$$

Notice that, since M and K are positive definite, there exists an invertible matrix Q , such that $Q^TMQ = I$ and $Q^TKQ = \Lambda$ for a diagonal matrix Λ (this is known as simultaneous diagonalization by congruence; see Corollary 1.7.18 in [11]). Therefore, letting $\tilde{a}_i = Q^{-1}a_i$, $\tilde{x}_i = Q^{-1}x_i$, and $\tilde{v}_i = Q^{-1}v_i$, we have

$$\begin{cases} \tilde{x}_{k+1} = \tilde{x}_k + h\tilde{v}_k + \frac{1}{2}h^2\tilde{a}_k \\ \tilde{v}_{k+1} = \tilde{v}_k + \frac{1}{2}h(\tilde{a}_k + \tilde{a}_{k+1}) \\ MQ\tilde{a}_k = -KQ\tilde{x}_k - KQ\beta h^2\tilde{a}_k \end{cases} \quad (16)$$

Multiplying the last equation by Q^T on the left, we get

$$\tilde{a}_k = -\Lambda\tilde{x}_k - \Lambda\beta h^2\tilde{a}_k \quad (17)$$

Therefore, one can assume without loss of generality that M is the identity and K is a diagonal matrix; since the following analysis applies to diagonal elements, we can, without loss of generality, further assume M and K to be scalars.

Now, from the 3rd equation in (15), we have

$$a_k = -(M + K\beta h^2)^{-1} K X_k \quad (18)$$

Substituting the above expression for a_k and a_{k+1} into the updating rule, we obtain

$$\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} I - \alpha h^2/2 & hI \\ \alpha h(\alpha h^2/4 - I) & I - \alpha h^2/2 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} \quad (19)$$

where $\alpha = (M + K\beta h^2)^{-1} K$. Algebraic computations show that the updating matrix in (19) has the following eigenvalues:

$$\lambda_{1,2} = \frac{1}{2}(2 - \alpha h^2 \pm h\sqrt{\alpha^2 h^2 - 4\alpha}) \quad (20)$$

It is not difficult to show that if $\alpha^2 h^2 - 4\alpha \leq 0$ then $\|\lambda_{1,2}\| = 1$ (and hence the trajectory remains bounded). Since $M > 0$ and $K > 0$, we only need to show that $\alpha h^2 - 4 \leq 0$. Because

$$\alpha h^2 = \frac{Kh^2}{M + K\beta h^2} \leq \frac{Kh^2}{K\beta h^2} = \frac{1}{\beta}, \quad (21)$$

if $\beta \geq 1/4$, the above is ≤ 4 . This finishes the proof. \square

Remark 2.2. This theorem can be easily understood: since only linear test problems are considered, SyLiPN is identical to Push-forward Newmark, which is equivalent to Newmark in terms of stability. The stability statement follows because Newmark is well-known to be unconditionally stable when $\beta \geq 1/4$.

Remark 2.3. For the special case of a stiff potential $V(x) = V_0(x) + \epsilon^{-1}V_1(x)$, the following modification of SyLiPN will also be unconditionally linearly stable as long as $\beta > 1/4 + \mathcal{O}(\epsilon)$:

Integrator 4. Simplified SyLiPN for stiff systems ($\epsilon \gg 1$):

$$\begin{cases} x_{k+1} &= x_k + hv_k + \frac{1}{2}h^2 a_k \\ v_{k+1} &= v_k + \frac{1}{2}h(a_k + a_{k+1}) \\ a_k &= -M^{-1}(\nabla V_0(x_k) + \epsilon^{-1}\nabla V_1(x_k)) - M^{-1}\epsilon^{-1}\text{Hess}V_1(x_k)\beta h^2 a_k \end{cases} \quad (22)$$

i.e., the Hessian of the small V_0 can be neglected. This is because assuming $V_0(x) = \frac{1}{2}x^T K_0 x$ and $V_1(x) = \frac{1}{2}x^T K_1 x$, then α in (19) will be

$$\alpha = \frac{K_0 + \epsilon^{-1}K_1}{M + \epsilon^{-1}K_1\beta h^2} \quad (23)$$

It can be shown that the stability condition $\alpha^2 h^2 - 4\alpha \leq 0$ is satisfied for all h as long as $\beta > 1/4 + \mathcal{O}(\epsilon)$, because (23) can be rewritten as

$$K_0 h^2 - 4M \leq \epsilon^{-1}K_1(4\beta - 1)h^2 \quad (24)$$

and if this were to violate, the only possibility is that the right hand side is small enough because $h = \mathcal{O}(\sqrt{\epsilon})$, but then the left hand side is < 0 , and hence this hypothetical violation will not happen.

As $\epsilon \rightarrow 0$, $\beta > 1/4$ will be enough to guarantee stability.

Remark 2.4. [12] uses an energy bound based on an assumption to demonstrate the nonlinear stability of Newmark. The same energy bound applies to SyLiPN, because Newmark and SyLiPN differ in force estimations by only a high order term of $\mathcal{O}(h^2)$, which will not affect the leading term in the energy bound. The assumption introduced there, however, can not be checked a priori for either Newmark or SyLiPN. Possible violations of this assumption may result in a nonlinear instability of Newmark or SyLiPN. As commented before, unconditional stability for arbitrary nonlinear systems is beyond the scope of current researches.

Remark 2.5. For possible improvements on nonlinear stability, one may resort to linearizations of more stable methods (such as those in [18]). However, few methods remain symplectic after the linearization.

Theorem 2.2 (Consistency). *Consider an integrator for (1) given by the following updating rule:*

$$\begin{cases} x_{k+1} &= x_k + hv_k + \frac{1}{2}h^2a_k \\ v_{k+1} &= v_k + \frac{1}{2}h(a_k + a_{k+1}) \\ a_k &= -M^{-1}\nabla V(x_k) - M^{-1}f(x_k)h^2a_k \end{cases}, \quad (25)$$

where $f \in \mathcal{C}(Q)$ is an arbitrary function.

If $V \in \mathcal{C}^3(Q)$, this integrator has a 3rd order local error, i.e., start with x_k, v_k at time kh , and denote by $\tilde{x}_{k+1}, \tilde{v}_{k+1}$ the exact solution at time $(k+1)h$ and by x_{k+1}, v_{k+1} the numerical solution after one-step update, then $\tilde{x}_{k+1} - x_{k+1} = \mathcal{O}(h^3)$ and $\tilde{v}_{k+1} - v_{k+1} = \mathcal{O}(h^3)$.

Proof. Assume without loss of generality that $M = I$. Writing $a(\cdot) = -\nabla V(\cdot)$, we have

$$a_k = a(x_k)/(1 + f(x_k)h^2) = a(x_k) + \mathcal{O}(h^2) \quad (26)$$

and

$$a(x_{k+1}) = a(x_k) + (x_{k+1} - x_k)a'(x_k) + \mathcal{O}((x_{k+1} - x_k)^2) = a(x_k) + hv_k a'(x_k) + \mathcal{O}(h^2) \quad (27)$$

Since the exact dynamics (1) is governed by

$$\begin{cases} \dot{x} = v \\ \dot{v} = a(x) \end{cases}, \quad (28)$$

if smoothness of the solution is assumed, we obtain by Taylor expansion that

$$\begin{aligned}
\tilde{x}_{k+1} &= x_k + hv_k + \frac{h^2}{2} \dot{v}_k + \mathcal{O}(h^3) \\
&= x_k + hv_k + \frac{h^2}{2} (a_k + \mathcal{O}(h^2)) + \mathcal{O}(h^3) \\
&= x_k + hv_k + \frac{h^2}{2} a_k + \mathcal{O}(h^3) = x_{k+1} + \mathcal{O}(h^3)
\end{aligned} \tag{29}$$

$$\begin{aligned}
\tilde{v}_{k+1} &= v_k + ha(x_k) + \frac{h^2}{2} a'(x_k) v_k + \mathcal{O}(h^3) \\
&= v_k + \frac{h}{2} (a(x_k) + a(x_k) + ha'(x_k) v_k) + \mathcal{O}(h^3) \\
&= v_k + \frac{h}{2} (a_k + \mathcal{O}(h^2) + a(x_{k+1}) + \mathcal{O}(h^2)) + \mathcal{O}(h^3) \\
&= v_k + \frac{h}{2} (a_k + \mathcal{O}(h^2) + a_{k+1} + \mathcal{O}(h^2)) + \mathcal{O}(h^3) = v_{k+1} + \mathcal{O}(h^3)
\end{aligned} \tag{30}$$

□

Corollary 2.1. *SyLiPN (Integrator 1) has a 3rd order local error.*

Proof. This is because SyLiPN (2) is a special case of (25) with $f(x_k) = \text{Hess}V(x_k)\beta$. □

Remark 2.6. By the celebrated Lax-Richtmyer equivalence theorem [20], a $\mathcal{O}(h^{p+1})$ local error (also known as ‘consistency’ when $p \geq 1$) together with stability will lead to a $\mathcal{O}(h^p)$ global error (i.e., ‘convergence’ for $p \geq 1$). For our case, if SyLiPN is stable, then it is 2nd order convergent (with a $\mathcal{O}(h^2)$ global error).

Theorem 2.3 (Symplecticity). *Consider an integrator given by the following updating rule:*

$$\begin{cases} x_{k+1} &= x_k + hv_k + \frac{1}{2}h^2 a_k \\ v_{k+1} &= v_k + \frac{1}{2}h(a_k + a_{k+1}), \\ a_k &= f(x_k) \end{cases} \tag{31}$$

where $f \in \mathcal{C}^1(Q)$ is an arbitrary function. This integrator is symplectic.

Proof. Compute the Jacobian of Eq.(31) (viewed as $x_k, v_k \mapsto x_{k+1}, v_{k+1}$):

$$\begin{aligned}
\mathcal{D} &:= \begin{bmatrix} \frac{\partial x_{k+1}}{\partial x_k} & \frac{\partial x_{k+1}}{\partial v_k} \\ \frac{\partial v_{k+1}}{\partial x_k} & \frac{\partial v_{k+1}}{\partial v_k} \end{bmatrix} \\
&= \begin{bmatrix} I + \frac{1}{2}h^2 f'(x_k) & hI \\ \frac{1}{2}h (f'(x_k) + (I + \frac{1}{2}h^2 f'(x_k)) f'(x_{k+1})) & I + \frac{1}{2}h^2 f'(x_{k+1}) \end{bmatrix}
\end{aligned} \tag{32}$$

Let $\mathcal{J} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$ be the canonical symplectic matrix. Basic algebra shows that

$$\mathcal{D}^T \mathcal{J} \mathcal{D} = \begin{bmatrix} 0 & A \\ B & C \end{bmatrix}, \text{ where}$$

$$\begin{cases} A = (I + \frac{1}{2}h^2 f'(x_k)) (I + \frac{1}{2}h^2 f'(x_{k+1})) - \frac{1}{2}h^2 (f'(x_k) + (I + \frac{1}{2}h^2 f'(x_k)) f'(x_{k+1})) = I \\ B = (I + \frac{1}{2}h^2 f'(x_k)) (-I - \frac{1}{2}h^2 f'(x_{k+1})) + \frac{1}{2}h^2 (f'(x_k) + (I + \frac{1}{2}h^2 f'(x_k)) f'(x_{k+1})) = -I \\ C = h(-I - \frac{1}{2}h^2 f'(x_{k+1})) + h(I + \frac{1}{2}h^2 f'(x_{k+1})) = 0 \end{cases}$$

which is indeed the definition of symplecticity $\mathcal{D}^T \mathcal{J} \mathcal{D} = \mathcal{J}$. \square

Corollary 2.2. *SyLiPN (Integrator 1) is symplectic.*

Proof. In SyLiPN, a_k is given by (3), which is a function of x_k . Therefore, SyLiPN is in the form of (31) and hence symplectic. \square

Remark 2.7. Linearizing Newmark or implicit midpoint will not result in a symplectic method, although the resulting integrator will be linearly-implicit, stable and 2nd-order. It is rare that the linearization of an implicit method is symplectic. See also Section 4.1 for a numerical demonstration on the consequence of loss of symplecticity.

On the other hand, Theorem 2.3 provides a family of symplectic integrators, but few choices of $f(\cdot)$ will result in convergent and stable ones.

SyLiPN lies in the intersection of linearized approaches and the family of symplectic integrators given by (31).

Remark 2.8. By Theorems 2.2 and 2.3, simplified SyLiPN for stiff systems (Integrator 4) is also 2nd-order convergent (if stable) and symplectic. It is also unconditionally linearly stable according to Remark 2.3.

3 Proof of Theorem 1.2

Consider the solution of (7). Due to energy conservation in the modified system, $\dot{q}M\dot{q}/2 + V(q) + \frac{1}{2}\omega^2 g(q)^T g(q)$ remains constant in time, and therefore $\frac{1}{2}\omega^2 g(q)^T g(q)$ remains bounded independently from ω as long as $V(\cdot)$ is bounded from below (a requirement that needs to be satisfied only in the domain containing the solution). As a consequence, $g(q)$ is at most $\mathcal{O}(1/\omega)$ and the constraint is satisfied approximately. In fact, $g(q)$ can be further shown to be of the order of $\mathcal{O}(1/\omega^2)$ in most cases (we refer to, for instance, [16], or Remark 3.1 and the multiscale analysis below).

Theorem 1.2 shows that the modified system (7) approximates the Lagrange multiplier approach represented by the DAE (6) and describe the associated topology. Observe that there are more than 2000 papers referring to the modified system, i.e., the penalty approach [33, 36, 26].

The proof of Theorem 1.2 is based on two observations: $g(q^\omega)$ is close to 0, and the two scale flow convergence (F-convergence) holds when the constrained manifold is flat. We first elaborate these facts by the following lemmas.

Lemma 3.1. *If the potential $V(\cdot)$ is bounded from below, then there exists a constant C , such that $\|g(q^\omega(s))\| \leq C/\omega$ for all s . Moreover, if $V(q)$ diverges towards infinity as $|q| \rightarrow \infty$, then the solution q^ω is bounded, i.e., there exists a constant \tilde{C} such that $\|q^\omega(s)\| \leq \tilde{C}$.*

Proof. Notice that the energy $[\dot{q}^\omega]^T M \dot{q}^\omega / 2 + V(q^\omega) + \omega^2 g(q^\omega)^T g(q^\omega)$ in the modified system (7) is conserved and bounded (due to the initial condition). Therefore, $V(\cdot)$ being bounded from below and $g^T g \geq 0$ imply that $\omega^2 g(q^\omega)^T g(q^\omega) = \mathcal{O}(1)$. Hence $g(q^\omega(s)) = \mathcal{O}(1/\omega)$.

By a similar energy argument, since $[\dot{q}^\omega]^T M \dot{q}^\omega / 2 \geq 0$ and $g(q^\omega)^T g(q^\omega) \geq 0$, $V(q^\omega)$ is bounded from above too, which implies that q^ω remains bounded. \square

Lemma 3.2. *Consider the solution to a conserved mechanical system*

$$\begin{cases} \ddot{x}^\omega = f_1(x^\omega, y^\omega) \\ \ddot{y}^\omega = f_2(x^\omega, y^\omega) - \omega^2 g(y^\omega)^T \nabla g(y^\omega) \end{cases} \quad (33)$$

where x^ω and y^ω are both vectors, and $x^\omega(0) = x_0$, $\dot{x}^\omega(0) = \dot{x}_0$, $y^\omega(0) = y_0$, $\dot{y}^\omega(0) = \dot{y}_0$. Suppose that f_1 , f_2 and ∇g are C^1 with bounded derivatives, x^ω and y^ω are bounded, $g(y_0) = 0$ and $\frac{d}{dt}g(y_0) = 0$, and $g(\cdot)$ has a non-degenerate Jacobian in a neighborhood of y_0 , then the limit

$$\lambda(t) := - \lim_{T \rightarrow 0} \lim_{\omega \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \omega^2 g(y^\omega(s)) ds \quad (34)$$

exists and is finite. Denote by $x(t)$, $y(t)$ the solution to

$$\begin{cases} \ddot{x} = f_1(x, y) \\ \ddot{y} = f_2(x, y) + \lambda^T \nabla g(y) \end{cases} \quad (35)$$

with the same initial conditions $x_0, \dot{x}_0, y_0, \dot{y}_0$, then as $\omega \rightarrow \infty$,

$$\begin{cases} x^\omega \rightarrow x \\ y^\omega \xrightarrow{F} y \\ g(y^\omega) \rightarrow 0 \end{cases} \quad (36)$$

Proof. We employ the multiscale averaging theory studied in [31] to demonstrate the convergence. Here the constrained manifold is a submanifold with any x and y satisfying $g(y) = 0$. x^ω is a slow variable and its evolution corresponds to the constrained dynamics. y^ω is a fast variable corresponding to a fluctuating deviation from the constrained manifold at a characteristic timescale of $\mathcal{O}(1/\omega)$, and it lies in the normal bundle of the constrained manifold.

First, consider the case in which $g(\cdot)$ is a linear function $g(y) = Cy^T$ (the affine case can be similarly treated by shifting y). The dynamics of y is governed by

$$\ddot{y}^\omega = f_2(x^\omega, y^\omega) - \omega^2 y^\omega C^T C \quad (37)$$

This is a forced harmonic oscillator, and its solution can be written as:

$$y^\omega(t) = \int_0^t f_2(x^\omega(s), y^\omega(s)) \sin(\omega \tilde{C}s) \tilde{C}^{-1} / \omega ds \quad (38)$$

where $\tilde{C} = \sqrt{C^T C}$ is the well-defined matrix square root, and matrix sin is defined either by Taylor expansion or diagonalization. Notice that there is no propagation of initial conditions, because the initial conditions satisfy the constraint and are therefore null.

It can be shown from (38) (for instance, by Lemma 3.8 in [32]; the idea is that an addition $1/\omega$ comes from the sin due to integration by parts) that $y^\omega(t)$ is $\mathcal{O}(\omega^{-2})$ at least up to $t = o(1)$, and y^ω is asymptotically periodic (because (37) is asymptotically linear) and hence locally ergodic (as $\omega \rightarrow \infty$).

Since y^ω is locally ergodic, (34) well-defines λ , and Theorem 1.2 in [31] guarantees that the effective equation for (37) is

$$\ddot{y} = f_2(x^\omega, y) + \lambda^T C, \quad (39)$$

in the sense that $y^\omega \xrightarrow{F} y$ and $x^\omega \rightarrow x$. Notice that the convergence on x is in the strong sense, i.e., $\lim_{\omega \rightarrow \infty} x^\omega(t) \rightarrow x(t)$ for all bounded $t > 0$. This is because x is purely slow, for which case, F-convergence implies strong convergence.

Now consider a fully nonlinear $g(\cdot)$ with a non-degenerate Jacobian. Lemma 3.1 gives that $g(y^\omega) = \mathcal{O}(1/\omega)$. Since $g(\cdot)$ has a non-degenerate Jacobian, and y^ω is in a bounded and therefore compact set, we also have $y^\omega - y_0 = \mathcal{O}(1/\omega)$. Consequently, the dynamics of y^ω actually approaches that of a forced oscillator (with equilibrium at y_0) at a $\mathcal{O}(1/\omega)$ timescale, because $g(\cdot)$ approaches a linear function (which is its first order Taylor expansion):

$$\begin{aligned} \ddot{y}^\omega &= f_2(x^\omega, y^\omega) - \omega^2 g(y^\omega)^T \nabla g(y^\omega) \\ &= f_2(x^\omega, y^\omega) - \omega^2 (\nabla g(y_0)(y^\omega - y_0)^T + \mathcal{O}(\omega^{-2}))^T (\nabla g(y_0) + \text{Hess } g(y_0)(y^\omega - y_0)^T + \mathcal{O}(\omega^{-2})) \\ &= f_2(x^\omega, y^\omega) - \omega^2 (y^\omega - y_0) \nabla g(y_0)^T \nabla g(y_0) + \mathcal{O}(1) \end{aligned}$$

where the nonlinearity $f_2 + \mathcal{O}(1)$ again only manifests as a slow force, which is asymptotically dominated by the linear term that leads to periodic oscillations. Analogous to the above linear case, y^ω is hence locally ergodic, the Lagrange multiplier λ is asymptotically well-defined, and the solution x^ω, y^ω F-converges to the effective solution x, y . \square

With these two lemmas, we can prove Theorem 1.2. Figure 1 visualizes the notations used in the proof to help understand the geometry.

Sketch of the proof of Theorem 1.2. Since $g(q^\omega)$ is at most $\mathcal{O}(1/\omega)$ (Lemma 3.1), q^ω is close to the constrained manifold $g^{-1}(0)$ in the sense that if we define for all t :

$$q_0(t) := \min_{q \in g^{-1}(0)} \|q - q^\omega(t)\| \quad (40)$$

then $q^\omega(t) - q_0(t) = \mathcal{O}(1/\omega)$. Indeed, given that ∇g has the maximum rank, $\nabla g(q_0(t))$ spans the normal section (i.e., the subspace perpendicular to the tangent subspace) of

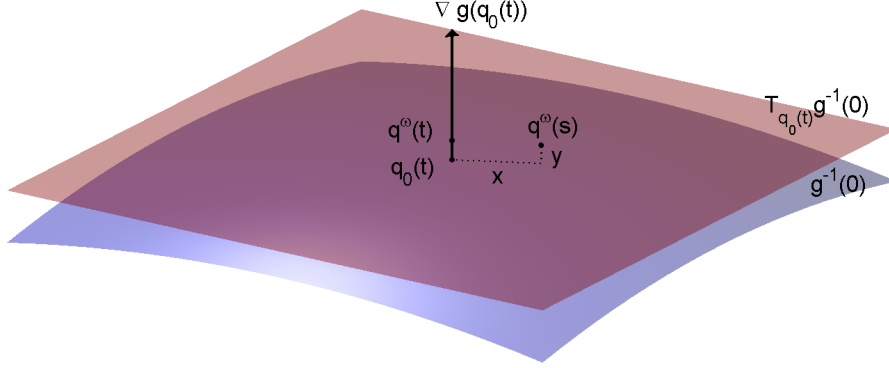


Figure 1: The multiscale geometry of the constrained dynamics – slow and fast variables are x and y .

the constrained manifold, in which $q^\omega(t) - q_0(t)$ also lies. Moreover, g restricted to each normal section is an isomorphism, and both the restricted map and its inverse have bounded norms due to the boundedness of q^ω (i.e., compactness of the solution space) — this is why $g(q^\omega) = \mathcal{O}(1/\omega)$ implies $q^\omega(t) - q_0(t) = \mathcal{O}(1/\omega)$.

The idea is that since q^ω is close enough, the constrained manifold can be locally viewed as a flat subspace, and F-convergence for this case has already been proved in Lemma 3.2. Mathematically speaking, there exists a linear isomorphism $A_{q_0(t)}$, such that

$$A_{q_0(t)}(q^\omega(t) - q_0(t)) = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (41)$$

where y is a vector with codimension of $g^{-1}(0)$ and 0 is a null vector with dimension of $g^{-1}(0)$.

For $q^\omega(s)$ with $s - t = \mathcal{O}(1/\omega)$, we will have a full-dimensional representation:

$$A_{q_0(t)}(q^\omega(s) - q_0(t)) = \begin{bmatrix} x \\ y \end{bmatrix} \quad (42)$$

and x and y will respectively be the slow and fast variables, representing the constrained dynamics and fluctuations away from the constrained manifold (analogous to Lemma 3.2). This is because

$$\begin{aligned} \frac{d^2}{ds^2} \begin{bmatrix} x \\ y \end{bmatrix} &= A_{q_0(t)}(-\nabla V(q^\omega(s)) - \omega^2 g(q^\omega(s)) \nabla g(q^\omega(s))) \\ &= \begin{bmatrix} f_1(x, y) \\ f_2(x, y) \end{bmatrix} + \begin{bmatrix} \mathcal{O}(1) \\ -\omega^2 \tilde{g}(y) \nabla \tilde{g}(y) + \mathcal{O}(1) \end{bmatrix}, \end{aligned} \quad (43)$$

where f_1 and f_2 are defined as $A_{q_0(t)}(-\nabla V(q^\omega(s)))$. The top $\mathcal{O}(1)$ on the right hand side of (43) is because $A_{q_0(t)}$ rotates the normal section to the y -direction, i.e.,

$$A_{q_0(t)} \nabla g(q^\omega(s)) = A_{q_0(t)}(\nabla g(q_0) + \mathcal{O}(1/\omega)) = \begin{bmatrix} 0 \\ * \end{bmatrix} + \mathcal{O}(1/\omega) \quad (44)$$

where $*$ is some non-zero expression, and certainly $\mathcal{O}(1/\omega) = \mathcal{O}(1)$.

The bottom on the right hand side of (43) can also be intuitively obtained by using an analogous geometric argument, together with the Taylor expansion.

Since (43) corresponds to the locally flat system (33), Lemma (3.2) proved the existence of an equivalent Lagrange multiplier as well as the F-convergence towards it. Moreover, (43) and the global dynamics near the curved constrained manifold (7) is linked via a coordinate transformation $q^\omega \mapsto A_{q_0}(q^\omega - q_0)$, which, naturally, is slowly varying as q_0 changes. Since averaging via F-convergence (Theorem 1.2 in [31]) still works if the slow and fast variables are images of the original variable under a slowly varying diffeomorphism, the global dynamics (7) is F-convergent to a solution of (9). Notice that $g(q(t)) = 0$ in (9) is automatically satisfied, because $\lim_{\omega \rightarrow \infty} g(q^\omega(t)) = 0$.

Lastly, the solution to (9) is also the solution to (6). This is by the existence and uniqueness of the solution to differential algebraic equations with initial conditions.

■

Remark 3.1. The above proofs show that $g(q^\omega(s))$ is not only $\mathcal{O}(\omega^{-1})$ but in fact $\mathcal{O}(\omega^{-2})$.

Remark 3.2. If $-\lim_{\omega \rightarrow \infty} \omega^2 g(q^\omega(t))$ exists, then (8) simplifies to

$$\lambda(t) = - \lim_{\omega \rightarrow \infty} \omega^2 g(q^\omega(t)), \quad (45)$$

and the solution of the modified system (7) will not only F-converge but also strongly converge to that of (9). Such a strong equivalent Lagrange multiplier, however, does not always exist, at least it is not observed in our numerical experiments nor in analytical investigations of linear systems (results not shown).

An equivalent Lagrange multiplier in the sense of averaging (8), as shown above, always exists. For numerical demonstrations, see Figure 3 (double pendulum) or Figure 10 (water cluster).

Remark 3.3. We only provided the main lines of the proof of Theorem 1.2. The details pose no difficulty, and are analogous to the analysis done in [31, 32]. We chose not to do so because full details would add 20 more pages to this paper and we do not want to diffuse the attention of the reader from the key contribution of this paper: SyLiPN's symplecticity (although the authors have not seen a mathematical justification of the modified mechanical system elsewhere, the modified system per se is, again, already widely acknowledged).

4 Examples

4.1 Double pendulum

Implementation: Consider a double pendulum system. One way to represent the system is to use 4 degrees of freedom and 2 nonlinear constraints in Euclidian coordinates.

Written in the above notations, we have

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 \\ 0 & 0 & m_2 & 0 \\ 0 & 0 & 0 & m_2 \end{bmatrix} \quad (46)$$

$$V(x_1, y_1, x_2, y_2) = -gy_1 - gy_2 \quad (47)$$

$$g(x_1, y_1, x_2, y_2) = \begin{bmatrix} x_1^2 + y_1^2 - L_1^2 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_2^2 \end{bmatrix} \quad (48)$$

where m_1, m_2 are two masses, g is the gravitational constant, and L_1, L_2 are lengths of the two pendulums. To simplify our notations, we adopt a dimensionless convention and assume $m_1 = m_2 = g = 1$.

Implementation of our constraint-free approach on this system is straightforward (Eq. 7).

To integrate in generalized coordinates θ, ϕ , let $x_1 = L_1 \sin \theta, y_1 = -L_1 \cos \theta, x_2 = L_1 \sin \theta + L_2 \sin \phi, y_2 = -L_1 \cos \theta - L_2 \cos \phi$. Then the Lagrangian $L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M \dot{q} - V(q)$ with $q = [x_1 \ y_1 \ x_2 \ y_2]$ simplifies to be

$$\tilde{L}(\theta, \phi, \dot{\theta}, \dot{\phi}) = \frac{1}{2} (2L_1^2 \dot{\theta}^2 + L_2^2 \dot{\phi}^2 + 2L_1 L_2 (\cos \theta \cos \phi + \sin \theta \sin \phi) \dot{\theta} \dot{\phi}) + 2L_1 \cos \theta + L_2 \cos \phi \quad (49)$$

in which the length constraints are intrinsically handled. Corresponding Euler-Lagrangian equations will give the constrained dynamics. Numerically, one approximates the action $\int_{kh}^{(k+1)h} \tilde{L}(\theta, \phi, \dot{\theta}, \dot{\phi}) dt$ using a quadrature rule and obtains a discrete Lagrangian $\tilde{L}_d(\theta_k, \phi_k, \theta_{k+1}, \phi_{k+1})$. Applying the least action principle again, a set of discrete Euler-Lagrangian equations [22] are obtained. Notice that the mass matrix in the generalized coordinates is

$$\tilde{M}(\theta, \phi) = \begin{bmatrix} 2L_1^2 & L_1 L_2 (\cos \theta \cos \phi + \sin \theta \sin \phi) \\ L_1 L_2 (\cos \theta \cos \phi + \sin \theta \sin \phi) & L_2^2 \end{bmatrix}, \quad (50)$$

which is no longer constant but position dependent. As a consequence, variational integrators given by discrete Euler-Lagrangian equations, even symplectic Euler or Verlet, will be implicit.

Regarding numerical approximations to the continuous Lagrange multiplier system (6), the well known algorithms of SHAKE and RATTLE can be viewed as variational integrators with 1st-order and 2nd-order quadrature discretizations of the Lagrange-d'Alembert principle $\int (L(x_1, y_1, x_2, y_2) + \lambda \cdot g(x_1, y_1, x_2, y_2)) dt$, in which constraints are realized via a vectorial Lagrange multiplier $\lambda(t)$ [22]. These methods also involve implicit solve at each step to compute the virtual force (λ) that reinforces the constraints.

Importantly, this system is much less trivial than it might appear, and the reason lies in the nonlinearity of the constraint g . In the modified system, for example, the additional constraining potential $\omega^2 g^T g / 2$ is not quadratic, and it results in cubically nonlinear ODEs.

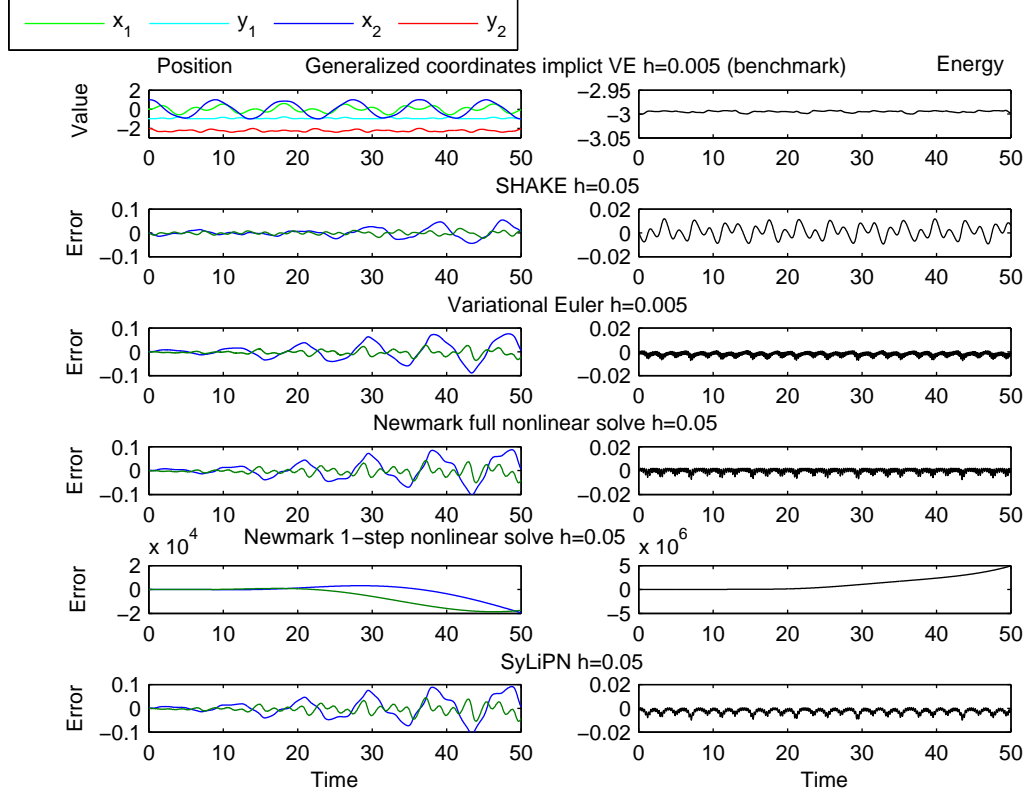


Figure 2: Integration errors by big step implicit SHAKE, small step explicit Variational Euler on the modified system (Eq. 7), big step implicit Newmark (with nonlinear systems fully solved at each timestep), big step linearly implicit Newmark (with the nonlinear system at each step solved to the first iteration — no longer symplectic), and big step SyLiPN. The benchmark is provided by small step Variational Euler in generalized coordinates (implicit). Initial positions are $x_1(0) = 0, y_1(0) = -1, x_2(0) = 1, y_2(0) = -2$ and initial momenta are zero, $L_1 = 1$ and $L_2 = \sqrt{2}$, and total simulation time is 100. The modified system (corresponding to Rows 3-6) uses $\omega = 20$. SyLiPN uses $\beta = 0.4$. Row 3 uses $h = 0.1/\omega$ for stability, and Row 2,4,5,6 use a 10x bigger $h = 0.05$. Position errors are only shown on x_2 and y_2 for readability.

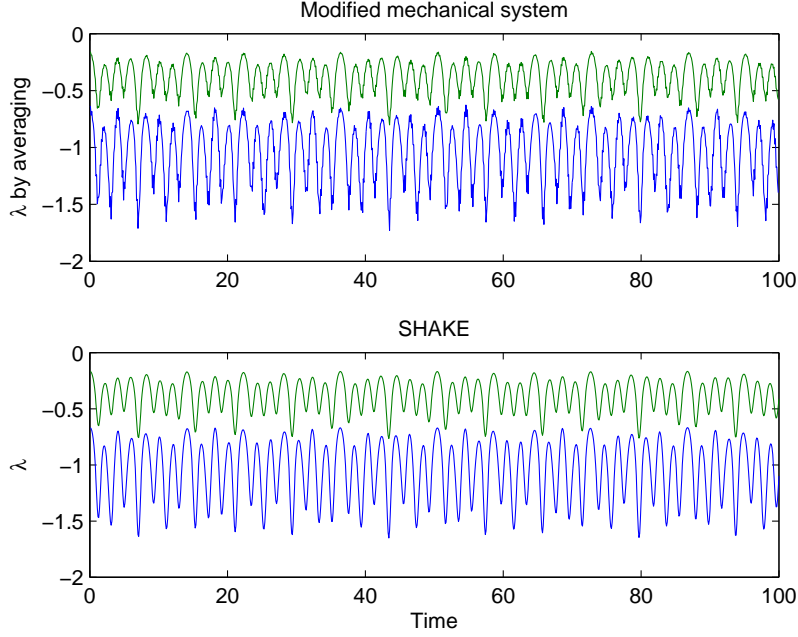


Figure 3: Comparison of Lagrange multipliers computed by (8) from the modified mechanical system (simulated by SyLiPN) and by SHAKE. Here Lagrange multiplier at each time is a 2-dim vector (indicated respectively by red and blue). All experimental parameters are the same as in Figure 2. The integral in (8) is numerically approximated by an empirical average at time points $t, t + 0.1/\omega, \dots, t + 0.2 - 0.1/\omega, t + 0.2$.

Results: Figure 2 provides an illustration of the numerical errors of different integration methods. Symplectic numerical methods based on the constraint-free modified system (Row 3,4,6) produce small errors comparable to that by standard iterative SHAKE (Row 2).

The original Newmark (Integrator 2) with only the first iteration taken when solving the nonlinear system (Row 5) also integrates the modified system, but its error is very big. This is because executing only the first Newton iteration is equivalent to linearizing the nonlinear update equations (analogous to how we obtain SyLiPN from Push-forward Newmark), which can be shown to be not symplectic any more. Since it can also be shown that linearized Newmark is a convergent method (i.e., with finite time simulation error arbitrarily small by choosing h small enough), its unsatisfactory performance is completely due to the lack of symplecticity — notice that the energy drift is drastic because the system is stiff. In the nonlinear solving perspective, this is equivalent to saying that the predictor from the last step does not give an accurate enough initial guess, and Newton solver has to take more than one iteration to solve the updating equation in order to keep the integration symplectic.

On the contrary, SyLiPN (Row 6) yields small numerical errors that are almost identical to those by the original Newmark (with the nonlinear system fully solved at

each timestep; Row 4). This is due to the nontrivial choice of Push-forward Newmark as the method to linearize, whose symplecticity survives the linearization. At the same time, it is not surprising that SyLiPN produces errors slightly larger than that by SHAKE, because when ω is finite, the modified system is just an approximation to the constrained dynamics, and this approximation induces additional error besides the integration error; Row 3 also indicates such an approximation error, and an additional discussion on it can be found in Section 4.2).

Figure 3 compares the Lagrange multiplier, computed by SHAKE with the equivalent Lagrange multiplier obtained from the modified mechanical system (7) via averaging (8). The agreement between their trajectories validates numerically the equivalency between the well-established Lagrange multiplier approach (6) and the modified system (7) (Theorem 1.2; notice that the constraint g is not an affine function but nonlinear).

Speed wise, generalized coordinate implicit VE (benchmark), SHAKE, Variational Euler, Newmark with full nonlinear solve, Newmark with one-step nonlinear solve, and SyLiPN respectively spent 77.9, 5.8, 0.8, 8.1, 0.3 (4.6 if HessV is not analytically provided but approximated by the nonlinear solver), and 0.3 seconds on the above simulation (on a 2.4GHz laptop running MATLAB 7.7 and ‘fsolve’ as the nonlinear solver with its default 10^{-6} termination tolerance). Except VE on the modified system, all methods use a large timestep independent of ω . Generalized coordinate approach, SHAKE, and Newmark are based on solving nonlinear equations, which significantly slowed down the computation. Linearized implicit integrators use a large step and only involve solving linear systems, and therefore are superior in terms of computational efficiency.

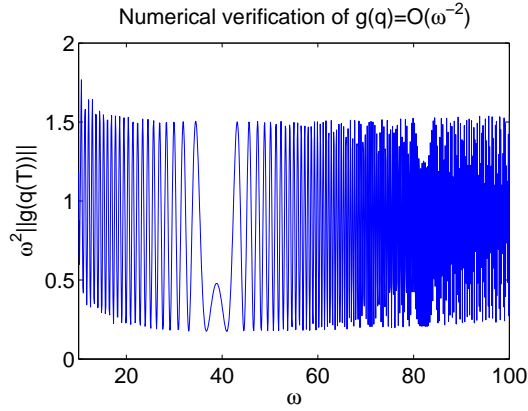


Figure 4: $\omega^2 \|g(q(T))\|$ numerically computed by SyLiPN as a function of ω . ω is enumerated from 10 to 100 with an increment of 0.1, $T = 50$ is fixed, and experimental parameters are the same as in Figure 2, except that a smaller timestep $h = 0.01$ is used for a more accurate demonstration (the timestep is still coarse, because Variational Euler or Velocity Verlet requires $h = 0.001$ for stability).

In addition, Figure 4 numerically shows that $\omega^2 \|g(q(T))\| = \mathcal{O}(1)$ by ω values, and therefore verifies that $g(q) = \mathcal{O}(1/\omega^2)$ (Remark 3.1).

Also worth mentioning is, given the same integration step size, a linearly-implicit method (e.g., SyLiPN) is not necessarily slower than an explicit one with complicated

force evaluations. In fact, we analytically pre-compute a position dependent nonlinear function, which corresponds to the inversion of the matrix $I + \text{Hess}V(x)\beta h^2$, so that SyLiPN (Integrator 1) becomes entirely explicit. Doing so, however, does not result in a gain in speed, because it is expensive to evaluate this pre-computed matrix function, which is then multiplied by a vector to accomplish the force evaluation. In fact, we counted the time elapses of one implementation based on linear solves and another based on explicit multiplications, and they respectively spent 0.3 and 4.2 seconds (the inverse matrix was generated automatically by Mathematica function ‘Inverse’, and therefore the code for computing it may not be optimized). In this sense, a linearly-implicit method is good enough.

4.2 Convergence test

By Corollary 2.1 and Remark 2.6, we know that SyLiPN is a 2nd order integrator. However, this is only a quantification of its integration ability at small h . In this subsection, we will numerically investigate two additional questions:

- (i) How does the integration error of SyLiPN scale with the timestep h when h is large?
- (ii) Besides the error of ODE integration, the constrained dynamics model (7) is also just an approximation. Combining both approximations, how does the error of SyLiPN simulation of the constrained dynamics depend on h ?

Double pendulum: First, continue our numerical experiments on the double pendulum system (Section 4.1). The same experimental setting will be used, but we decrease the simulation time from 50 to 10 so that the following investigation can be done within a short time. In addition, we further decrease h from 0.005 to 0.0001 in the generalized coordinate implicit VE so that the benchmark is more accurate.

To investigate the convergence rate, we compute $\log_\alpha \text{error}(\alpha h) / \text{error}(h)$ as a function of h , where $\text{error}(h)$ is the numerical integration error when the integration step is h , and α is a constant close to 1 (we chose $\alpha = 0.9$). This should converge to 2 for a second-order method. More generally, assuming

$$\text{error}(h) = Ch^p \tag{51}$$

for some constants $C > 0$ and p , it immediately follows that

$$p = \log_\alpha \text{error}(\alpha h) / \text{error}(h) \tag{52}$$

To study question (i), we obtain $\text{error}(h)$ by comparing the SyLiPN integration of (7) with timestep h and a benchmark integration of the same equation obtained by Velocity Verlet with a very small timestep $0.0001/\omega$. As can be seen from Figure 5, the convergence rate starts with being faster than cubic when h is macroscopic (independent of $1/\omega$), but eventually approaches being quadratic when h is microscopic (i.e., $h \ll 1/\omega$, $1/\omega = 0.05$ in our case), which is consistent with Corollary 2.1. Moreover, peculiar behavior exhibits in the mesoscopic region (i.e., $h \approx 1/\omega$), where the convergence rate

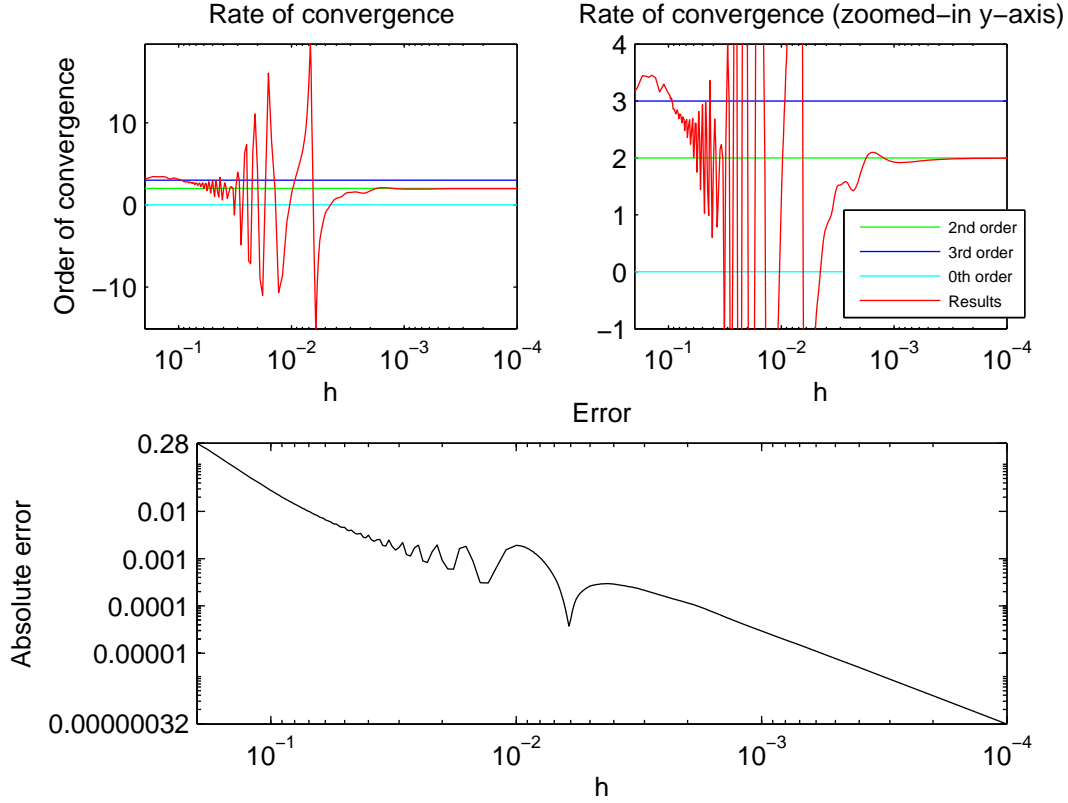


Figure 5: Convergence rate and integration error of SyLiPN as functions of h . h is enumerated as 0.0001, 0.0002, \dots , 0.0099, 0.01, 0.011, \dots , 0.099, 0.1, 0.11, \dots , 0.2.

frequently changes back and forth between large positives and small negatives. This phenomenon, together with the super-quadratic fast convergence at macroscopic h , explains why the error does not blow up when $h \gg 1/\omega$ and agrees with the stability of SyLiPN.

To investigate question (ii), we carry out a similar analysis, but this time the benchmark is obtained by generalized coordinate implicit Variational Euler with a very small step $h = 0.0001$. As can be seen in Figure 6, the convergence rate this time decreases from super-quadratic (at macroscopic h) to 0 (i.e., non-converging, at microscopic h). This is because the modified system (7) introduces an $\mathcal{O}(1/\omega)$ error from the true constrained dynamics (5). This error could be reduced by increasing ω (plot not shown), but a very large ω (and a corresponding small h ; also notice that the true parameter in the equation is ω^2) is rarely necessary, unless a high precision beyond 1/100 is in demand.

In fact, similar h -dependence of the convergence rate is observed when $\omega = 100$, but we will not repeat the results. We chose to show plots for $\omega = 20$ because this is the previously used value (in Section 4.1).

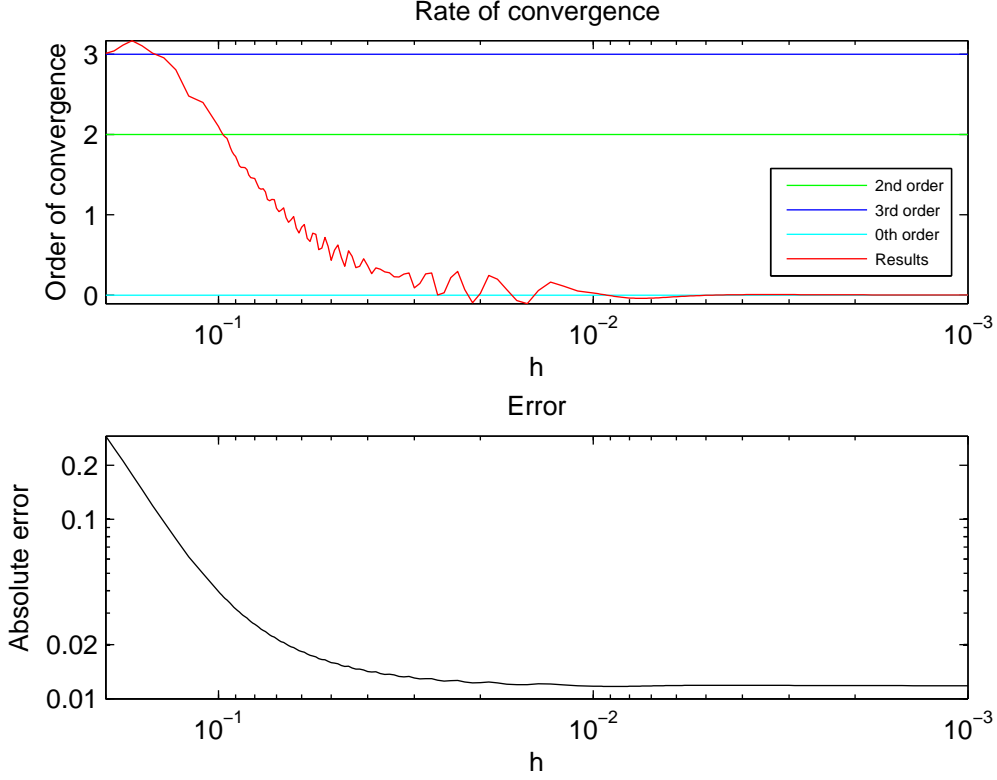


Figure 6: Convergence rate and error of SyLiPN when compared to the constrained dynamics. h is enumerated as 0.001, 0.0011, \dots , 0.0099, 0.01, 0.011, \dots , 0.099, 0.1, 0.11, \dots , 0.2.

Moreover, we see in Figure 7 that the numerical error of SHAKE actually grows back bigger and then stabilizes as h is further decreased to very small values. At the same time, SyLiPN is not convergent to the benchmark either, but its error is one-order-of-magnitude smaller. This non-vanishing error of SHAKE is due to the inevitable numerical errors in solving nonlinear systems: we plotted the Lagrange multiplier λ in SHAKE as a function of time (results not shown) and found that λ was 0 in the first few steps but then suddenly became very large; this is because h is so small that in the first few SHAKE steps the nonlinear solver thinks the constraints are satisfied without the need for any virtue force, but this leads to that the constraints are more and more poorly satisfied, and eventually a large Lagrange multiplier suddenly has to be chosen, which results in a non-zero numerical integration error.

One might tend to attribute this issue of SHAKE to an inappropriate choice of the termination tolerance used by the nonlinear solver; this, however, is not true. In fact, no matter how small the termination tolerance is, there always exist a small enough h_0 , such that the numerical error of SHAKE will not be further reduced by choosing $h < h_0$.

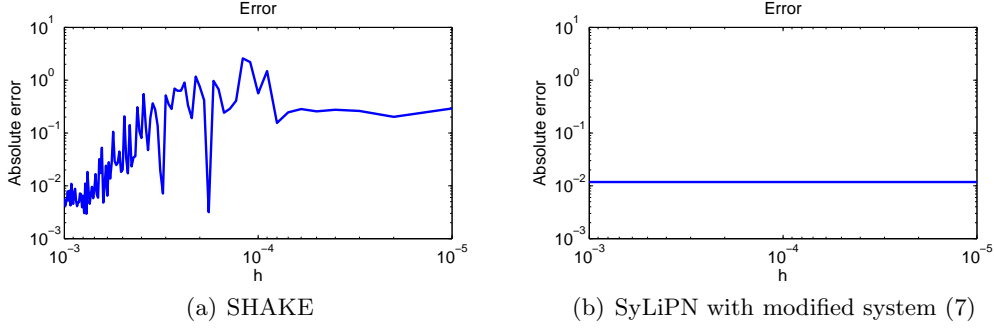


Figure 7: Errors of SHAKE and SyLiPN simulations of the constrained dynamics as functions of h . h is chosen from 10^{-5} to 10^{-3} with an increment of 10^{-5} . There is no ω involved in SHAKE, but for comparison with SyLiPN we mention that $10^{-5} \ll h_0/\omega$ and $10^{-3} \lesssim h_0/\omega$.

For instance, Figure 7 was obtained by using ‘fsolve’ in MATLAB as the nonlinear solver with termination tolerances on function value and variable both 10^{-6} (its default); if we decrease these tolerances to smaller values, for instance 10^{-7} , smaller h ’s (e.g., 10^{-7} as we tried) will make the final time integration error big. Some other may suspect the inaccuracy of the benchmark, but this is not the reason either (see below).

Uniform circular motion: To rule out the possibility of an inaccurate benchmark, we repeated the above experiments on a particle with no potential energy but constrained by $x^2 + y^2 = 1$, whose dynamics can be solved exactly and corresponds to a uniform circular motion. We obtained similar results (with slightly different numerical values of powers), and SHAKE still exhibits inaccuracy at tiny h , but we will not repeat the details.

4.3 High dimensional case: a chain of many pendulums

Now consider a high dimensional generalization: a chain of finitely many pendulums (which approximates a rope, except ropes in reality are subject to dissipations and

therefore not chaotic). The system is similarly modeled by Eq. 7 with:

$$M = \begin{bmatrix} m_1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & m_1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \vdots \\ \vdots & \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & m_n & 0 \\ 0 & 0 & \cdots & \cdots & 0 & m_n \end{bmatrix} \quad (53)$$

$$V(x_1, y_1, \dots, x_n, y_n) = - \sum_{i=1}^n g y_i \quad (54)$$

$$g(x_1, y_1, \dots, x_n, y_n) = \begin{bmatrix} x_1^2 + y_1^2 - L_1^2 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_2^2 \\ \vdots \\ (x_n - x_{n-1})^2 + (y_n - y_{n-1})^2 - L_n^2 \end{bmatrix} \quad (55)$$

where n indicates the total number of pendulums. We again assume without loss of generality that $m_i = 1$ and $g = 1$.

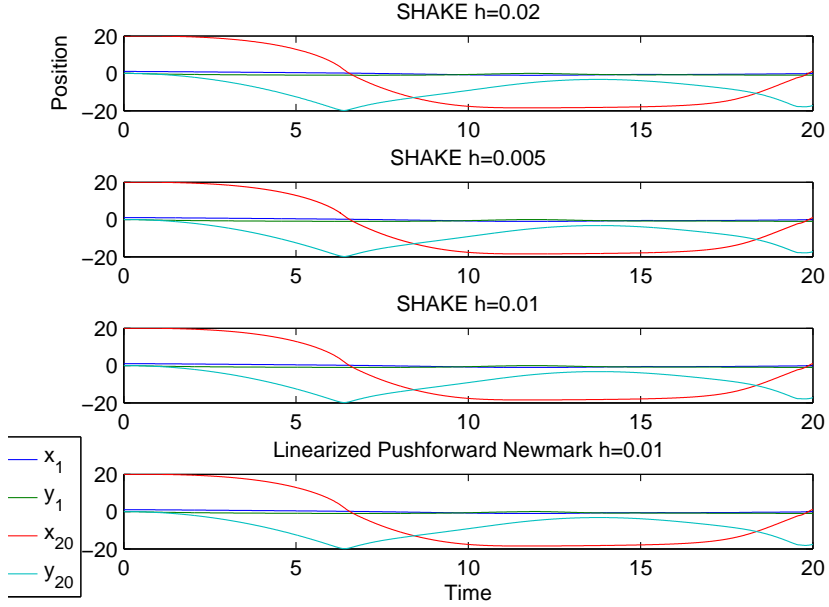


Figure 8: Comparison of $x_1(t), y_1(t), x_n(t), y_n(t)$ integrated by implicit SHAKE with $h = 0.005$, $h = 0.01$ and $h = 0.02$ and linearly-implicit SyLiPN with $h = 0.01$ on the modified system (Eq. 7). $n = 20$; initial positions are $x_i(0) = i, y_i(0) = 0$ for $i = 1, 2, \dots, n$ and initial momenta are zero; $L_i = 1$; the modified system uses $\omega = 100$; total simulation time is 20; SyLiPN uses $\beta = 0.4$.

Figure 8 provides a comparison between SHAKE and SyLiPN. The system is chaotic,

meaning that even the same converging integrator with slightly different timestep lengths will eventually produce completely different trajectories, and therefore we terminate the integration before chaotic behavior starts to manifest so that the comparison still makes sense. Such a termination time is decided so that SHAKE with different integration step lengths produce the same trajectory, but simulations beyond that time will yield significant deviations. SyLiPN agrees well with SHAKE till this termination time. Notice that we only present trajectories in Figure 8 but not integration errors. This is because a benchmark cannot be provided due to the chaotic nature of the system and the lack of an analytical solution; furthermore, the generalized coordinate approach is not applicable any more, because the mass matrix (high dimensional version of (50)) will be dense and complicated, and corresponding nonlinear updating equations will be very difficult to solve.

Speed wise, SHAKE with $h = 0.01$, $h = 0.005$, $h = 0.02$ and SyLiPN with $h = 0.01$ respectively spent 18.9, 37.1, 11.5 and 1.5 seconds on the above simulation (on a 2.4GHz laptop running MATLAB 7.7 and ‘fsolve’ as the nonlinear solver with its default 10^{-6} termination tolerance). Again, SyLiPN based on linear solves demonstrates a clear speed advantage.

4.4 Molecular dynamics of water cluster

We also use our method to simulate molecular dynamics. The demonstration example here is a cluster of water molecules, each interacting with others via non-bonded interactions of electrostatic and van der Waals forces, both of which are highly nonlinear.

Water model: The detailed potential that we use is the one in the prevailing TIP3P model (see for instance [14]). A system with N molecules has $9N$ degrees of freedom, because each water molecule consists of one oxygen atom and two hydrogen atoms, and each atom in 3D space is represented by 3 degrees of freedom. Indicating the a th molecule’s i th atom by position q_{ai} and momentum p_{ai} (each is three-dimensional), we write the Hamiltonian of the system as:

$$\mathcal{H} = \sum_{a=1}^N \sum_{i=1}^3 \frac{1}{2} p_{ai}^T m_i^{-1} p_{ai} + \sum_{a=1}^{N-1} \sum_{b=a+1}^N \left(\sum_{i=1}^3 \sum_{j=1}^3 \frac{K_c Q_i Q_j}{r_{ai,bj}} + \frac{A}{r_{a2,b2}^{12}} - \frac{C}{r_{a2,b2}^6} \right) \quad (56)$$

where $r_{ai,bj} := \|q_{ai} - q_{bj}\|$ is the inter-atom distance, $m_1 = m_3$ correspond to the mass of the hydrogen atom and m_2 the oxygen atom, K_c is the electrostatic constant, Q_i is the partial charge of atom i relative to the charge of the electron, A and C are Lennard-Jones constants that together approximate van der Waals forces.

In this TIP3P model (as well as most other prevailing models such as SPC, BF, TIPS2, and TIP4P, which are discussed in, e.g., [14], [34], or references therein), each water molecule is considered as a rigid body, with lengths of the two O-H bonds and the H-O-H bond angle fixed as constants r_{OH} and α_{HOH} . Detailed values of these model parameters could be found in, for instance, [14].

Therefore, the following vectorial constraint ($3N$ -dimensional) will enforce the geometry of the water molecule:

$$g(q) = \begin{bmatrix} (q_{11} - q_{12})(q_{11} - q_{12})^T - r_{OH}^2 \\ (q_{13} - q_{12})(q_{13} - q_{12})^T - r_{OH}^2 \\ (q_{11} - q_{13})(q_{11} - q_{13})^T - r_{HH}^2 \\ \vdots \\ (q_{N1} - q_{N2})(q_{N1} - q_{N2})^T - r_{OH}^2 \\ (q_{N3} - q_{N2})(q_{N3} - q_{N2})^T - r_{OH}^2 \\ (q_{N1} - q_{N3})(q_{N1} - q_{N3})^T - r_{HH}^2 \end{bmatrix}, \quad (57)$$

where $r_{HH} := 2r_{OH} \sin(\alpha_{HOH}/2)$ is again a constant. Consequently, the modified system (which SyLiPN simulates) is modeled by the following Hamiltonian:

$$\tilde{\mathcal{H}} = \mathcal{H} + \frac{1}{2}\omega^2 \sum_{a=1}^N ((r_{a1,a2}^2 - r_{OH}^2)^2 + (r_{a3,a2}^2 - r_{OH}^2)^2 + (r_{a1,a3}^2 - r_{HH}^2)^2) \quad (58)$$

Notice that the constrained problem is highly nontrivial (even though we already recasted the bond angel constraint as an equivalent constraint on the H-H distance), because the penalty terms are quartic nonlinear functions but not quadratic.

Constant temperature simulations: A water cluster at a constant temperature is the most frequently simulated case, because (1) thermal fluctuations are indispensable components when modeling the reality, and (2) the N-body system is chaotic in nature, and therefore its long-time deterministic simulation has limited applicability. Based on a purely demonstrational purpose (this might not be the best model), we adopt a well-established approach of Langevin (see for instance [28]) to simulate the water cluster in a constant temperature environment. In this model, the molecules are perturbed by environmental noise, and such perturbations gradually balance with the internal dissipations so that thermal equilibrium is eventually achieved. Mathematically, we integrate the following Langevin stochastic differential equations that model the stochastic version of the modified mechanical system:

$$\begin{cases} dq = \frac{\partial \tilde{\mathcal{H}}}{\partial p} dt \\ dp = -\frac{\partial \tilde{\mathcal{H}}}{\partial q} dt - \gamma \frac{\partial \tilde{\mathcal{H}}}{\partial p} dt + \sqrt{2\gamma\beta^{-1}} dW \end{cases} \quad (59)$$

where W is a $9N$ -dimensional Brownian motion, β^{-1} is the constant temperature, and γ is a positive constant indicating the strength of the dissipation. The system admits an invariant measure, which is the constant temperature Boltzmann-Gibbs distribution (BG for short; also known as the canonical ensemble), given by:

$$\pi(q, p) = Z^{-1} \exp(-\beta \tilde{\mathcal{H}}), \quad (60)$$

where $Z = \int_{\mathbb{R}}^{18N} \exp(-\beta \tilde{\mathcal{H}}) dq dp$ is the partition function.

To simulate (59), we use the approach of Geometric Langevin Algorithm (GLA for short; see [4]), for GLA is shown to produce trajectories that are not only path-wise accurate but also convergent to BG (60) under reasonable assumptions. Specifically, we compose the one-step update of SyLiPN (1) with the exact flow map of an Ornstein-Uhlenbeck process (OU for short, given by $dp = -\gamma M^{-1}p dt + \sqrt{2\gamma\beta^{-1}}dW$) to obtain a stochastic version of SyLiPN. One link to the classical methods is that if a first-order approximation of the OU flow map is used instead of the exact flow, the composition will result in the well-known Euler-Maruyama scheme.

Similarly, we compose the SHAKE one-step update with the OU flow map to extend SHAKE to the constant temperature setting. Under some technical conditions, the resulting method will approximately sample a constrained BG distribution (see Remark 2.1 in [4]):

$$\hat{\pi}(q, p) = \hat{Z}^{-1} \exp(-\beta H), \quad (61)$$

where $\hat{Z} = \int_{T^*g^{-1}(0)} \exp(-\beta \tilde{\mathcal{H}}) dq dp$ is a different partition function, since the phase space is now the cotangent bundle $T^*g^{-1}(0)$ with its base being the constrained manifold.

One may ask about the relationship between (60) and (61). Numerical experiments suggest that (60) converges to (61) as $\omega \rightarrow \infty$ (see the below experiment). Analytically, we know the equivalency between each individual constant- H -orbits (see Theorem 1.2). However, a full answer to this interesting yet difficult question (the proof of Theorem 1.2 is already highly-nontrivial, let alone its thermodynamics extension) is beyond the scope of this paper, which focuses on the symplecticity of SyLiPN.

On the choice of model: As mentioned above, there are many successful water models besides TIP3P. In addition, there are many other popular settings in which water molecules are coupled with the environment in different ways, including the constant temperature and constant pressure scenario (considered, for instance, in [3]). However, since the aim of this paper is to solve general problems (not limited to molecular dynamics), this section is more for an integrator-demonstration purpose rather than to claim any expertise in molecular dynamics; this purpose, we hope, will be well served as long as SyLiPN and classical constrained dynamics integrators produce agreeable results on nontrivial models, and therefore we did not focus on a best choice of model.

Numerical results: In water simulations, one is often interested in the statistical distribution of interatomic oxygen-oxygen distances in the thermal equilibrium limit, also known as the OO radial distribution [14]. Figure 9 shows the histograms that approximate such a distribution, respectively obtained by long time simulations of SHAKE and SyLiPN. Here we chose a system with appropriately many water molecules ($N = 7, 63$ degrees of freedom), so that peaks in the distribution could be clearly discerned. Since both the locations and sizes of all the peaks agree very well between SHAKE and SyLiPN, this is a strong numerical evidence that SyLiPN works for this complicated constrained stochastic system. In order to get this result, SHAKE spent 13471.75 seconds, including 12283.50 seconds on solving nonlinear systems (with 10^{-6} termination

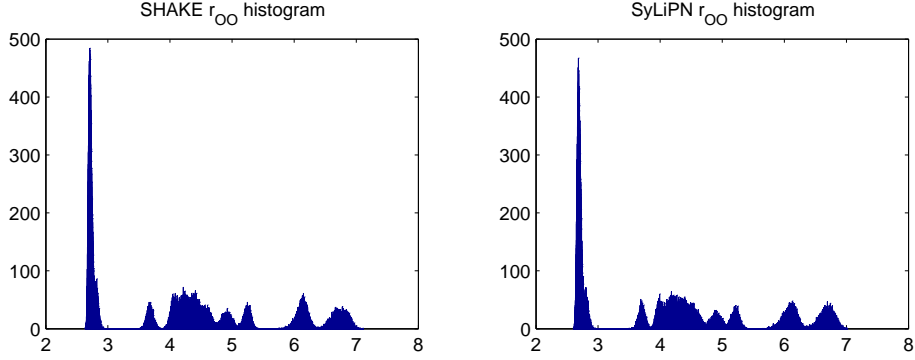


Figure 9: Empirical OO radial distribution in 7-water cluster obtained by long time ($T = 10000$) simulations of SHAKE and SyLiPN.

tolerance on variable and 10^{-10} termination tolerance on function value — the latter smaller because the constraint is high-dimensional and has a big Jacobian), whereas SyLiPN spent 1549.12 seconds, including 66.86 seconds on solving linear systems. $\sim 9x$ speed-up is obtained by SyLiPN.

Regarding experimental details, the initial configuration q_0 is obtained by minimizing the potential energy function $V(\cdot)$ starting with a random configuration (with $V > 1000$) via BFGS quasi-Newton’s method (see for instance [25]). However, we manually terminated optimization much earlier before it converges to a local minimum ($V(q_0) \approx -24$ and the local minimum corresponds to $V \approx -57$) — this way, we make sure that the convergence towards BG is totally due to the (correct) performance of SyLiPN and SHAKE but not an initial condition with high BG density. This q_0 is available at http://www.cds.caltech.edu/~mtao/Papers/Water7_qIC.txt. The initial momenta are all zero. Both SHAKE and SyLiPN simulate one trajectory of the system till time 10000. Judging from when the energy stops decreasing and instead starts to oscillate, the system equilibrates around time ~ 1000 (results not shown). We obtain the histograms of oxygen-oxygen distances by q ’s between time 5000 and 10000, and since this is much later than the mixing time, accuracy in approximating the distribution will be guaranteed due to ergodicity and law of large numbers. Histograms are drawn using 5000 bins so that the distribution profiles are smooth enough. The modified system uses $\omega = 20$. Both SHAKE and SyLiPN use an integration step $h = 0.05$ (notice that GLA based on Velocity Verlet requires a step 10x smaller for a stable integration of the modified system). Dissipation coefficient $\gamma = 0.01$, and the temperature inverse is $\beta = 50$.

We also use this example to provide another numerical demonstration of Theorem 1.2 — Figure 10 compares the Lagrange multiplier computed by SHAKE with the equivalent Lagrange multiplier obtained from the modified mechanical system (7) via averaging (8). These multipliers again agree very well (we turned off the noise and friction here; otherwise stochasticity will lead to different results and void the comparison).

Although the above demonstrations already involve 63 degrees of freedom, we now further increase N to 100 (900 degrees of freedom) to demonstrate the method’s ap-

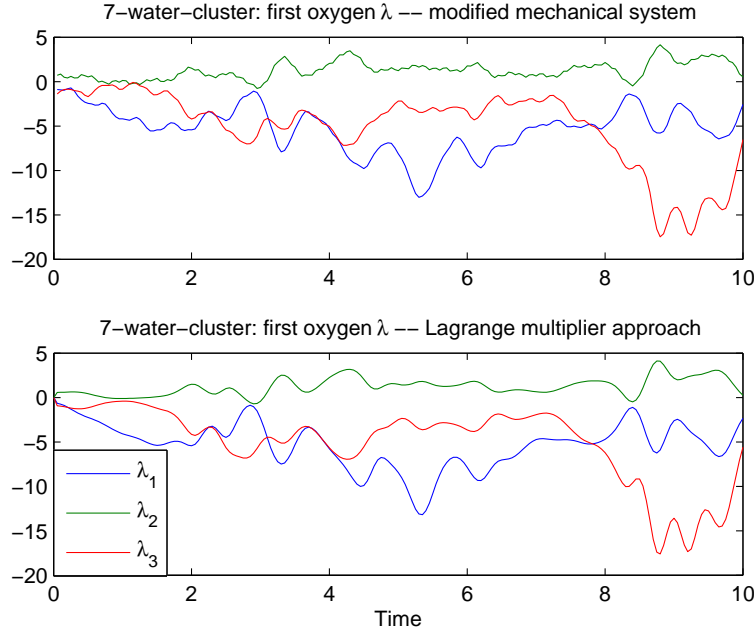


Figure 10: Comparison of Lagrange multipliers computed by (8) from the modified mechanical system (simulated by SyLiPN) and by SHAKE on water example. Only the Lagrange multiplier corresponding to the first oxygen atom is shown (3-dimensional vector). The integral in (8) is numerically approximated by an empirical average at time points $t, t + 0.1/\omega, \dots, t + 0.2 - 0.1/\omega, t + 0.2$. Due to the complicated dynamics of λ, ω is decreased to 500 so that visually no intrinsic details will be wiped out by the empirical averaging (notice that after all the equivalency is only an $\omega \rightarrow \infty$ asymptotic result). The demonstration is not over long time because numerical errors accumulate rapidly in deterministic simulations of a chaotic system.

plicability to large-scale problems. Figure 11 shows histograms that approximate the OO radial distribution in such a large system. The location and size of the first peak ($r_{OO} \approx 2.7$) respectively obtained by SHAKE and SyLiPN again agree well. Subsequent peaks, which are abundant, are however not easy to compare, because the simulation time is not long enough (although it is past the equilibration time) and hence there are not enough sample points to generate the empirical distributions with a small enough variance. Nevertheless, it is rather clear that SyLiPN produces results that agree with SHAKE.

In this experiment, SHAKE spent 42234.41 seconds, including 14272.05 seconds on solving nonlinear systems, and ~ 28000 seconds on force evaluations (the computations of V and ∇V), whereas SyLiPN spent 30059.47 seconds, including 318.98 seconds on solving linear systems, and ~ 29000 seconds on force evaluations (the computations of V , ∇V and Hess V). Notice that, although SyLiPN still saves a lot due to the linearization, most of the computation has been spent on force evaluations, and therefore SyLiPN does not improve the simulation efficiency as significantly as before. However, many brilliant approaches have been proposed to greatly accelerate force evaluations for molecular

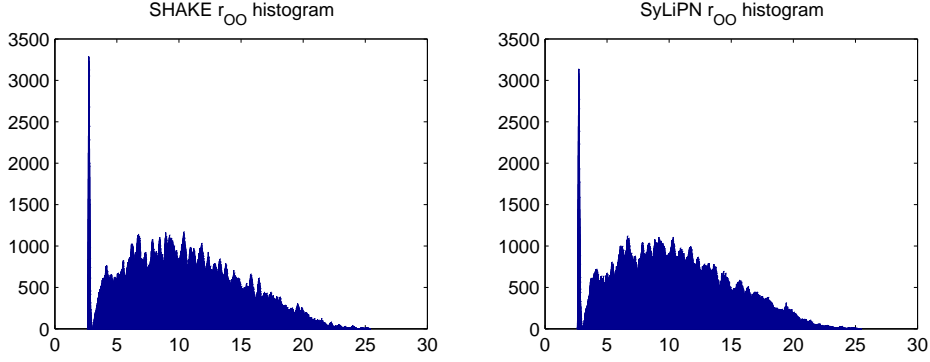


Figure 11: Empirical OO radial distribution in 100-water cluster obtained by simulations of SHAKE and SyLiPN till $T = 1000$.

dynamics, including the fast multipole method [8], or simply the idea of adopting a cut-off distance for ignoring weak long-range forces. Nevertheless, since the topic of force evaluations is parallel to designing integrators, we did not employ any of these sophisticated methods in our simulations, but only comment that they can greatly reduce the cost of force evaluation, hence render the acceleration by SyLiPN more significant in large systems.

It is important to notice that the times spent on force evaluations by SHAKE and SyLiPN are comparable; in other words, the computation of the Hessian in SyLiPN does not incur much computational overhead. This is because the potential is a function of relative distances, schematically $r_{ij} = \|x_i - x_j\|$. Notice that for any f

$$\frac{\partial^2 f(r)}{\partial x_i \partial x_j} = \frac{\partial r}{\partial x_i} \frac{\partial^2 f}{\partial r^2} \frac{\partial r}{\partial x_j} + \frac{\partial f}{\partial r} \frac{\partial^2 r}{\partial x_i \partial x_j}, \quad (62)$$

but $\frac{\partial r}{\partial x}$ and $\frac{\partial f}{\partial r}$ are already computed while obtaining the gradient, and $\frac{\partial^2 f}{\partial r^2}$ and $\frac{\partial^2 r}{\partial x_i \partial x_j}$ are not more expensive to compute. Furthermore, if a gradient method (such as Newton) is used as the nonlinear solver in Lagrange multiplier or generalized coordinate methods, the Hessian of the potential is either required or numerically estimated by the nonlinear solver, because the equation itself involves the gradient of the potential. In this sense, the computation of the Hessian is not even an overhead, unless a nonlinear solver that requires no gradient information is employed.

Not only that, the linear system associated with the Hessian can also be solved efficiently. For water clusters, the Hessian is in fact strongly dominated by 9-by-9 diagonal blocks (there are N of them). This is because the stiff penalty terms in (58) are localized, i.e., although there are $(9N)(9N+1)/2$ possible pairwise interactions, only $\mathcal{O}(N)$ of them are stiff. Therefore, solving a linear system with coefficient matrix $M + \text{Hess}V(x_k)\beta h^2$ can be done in $\mathcal{O}(N)$ time. This will be the usual case for polymers, unless the number of bonds is not at the order of the number of atoms. Moreover, simplified SyLiPN for stiff systems (Integrator 4, where the stiffness ϵ^{-1} is identified with ω^2) can be used for large-scale problems, and we actually use it in the above $N = 100$ water simulation

— after all, this simplified version is also convergent, symplectic, and stable. In simplified SyLiPN simulations of water clusters, the Hessian is completely block-diagonal, and the linear solve is executed molecule by molecule. Similarly, the nonlinear solve in SHAKE for enforcing the geometry of each water molecule was also carried out molecule by molecule; in other words, we have already used the optimal nonlinear solver in our implementation of SHAKE.

In addition, in case that one is reluctant to code up the Hessian evaluation (which will not be difficult if the potential is a function of relative distances), s/he also has the choice of approximating it by finite difference and $DOF + 1$ evaluations of ∇V . Due to Theorems 2.2 and 2.3, such an approximation will affect neither the convergence nor the symplecticity.

5 Acknowledgement

This work is supported by NSF grant CMMI-092600 and a generous gift from UTRC. We thank Mathieu Desbrun for the motivation, inspiring discussion and suggestions, and Eitan Grinspun for inspiring comments. We thank Carmen Sirois for proofreading the manuscript. We also thank several anonymous referees for constructive comments.

References

- [1] H. C. ANDERSEN, *Rattle: A “velocity” version of the shake algorithm for molecular dynamics calculations*, J. Comput. Phys, 52 (1983), pp. 24–34.
- [2] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users’ Guide*, SIAM, 3rd ed., 1999.
- [3] H. J. C. BERENDSEN, J. P. M. POSTMA, W. F. VAN GUNSTEREN, A. DINOLA, AND J. R. HAAK, *Molecular dynamics with coupling to an external bath*, J. Chem. Phys., 81 (1984), p. 3684.
- [4] N. BOU-RABEE AND H. OWHADI, *Long-run accuracy of variational integrators in the stochastic context*, SIAM J. Numer. Anal., 48 (2010), pp. 278–297.
- [5] F. CHIBA AND T. KAKO, *Newmark’s method and discrete energy applied to resistive mhd equation*, Vietnam J. Math., 30 (2002), pp. 501–520.
- [6] S. ERLICHER, L. BONAVENTURA, AND O. S. BURSI, *The analysis of the Generalized α method for non-linear dynamic problems*, Comput. Mech., 28 (2002), pp. 83–104.
- [7] M. FIXMAN, *Classical statistical mechanics of constraints: A theorem and application to polymers*, Proc. Nat. Acad. Sci. USA, 71-8 (1974), pp. 3050–3053.

- [8] L. F. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys, 73 (1987), pp. 325–348.
- [9] E. HAIRER, C. LUBICH, AND G. WANNER, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, Heidelberg Germany, second ed., 2004.
- [10] B. HESS, H. BEKKER, H. J. C. BERENDSEN, AND J. G. E. M. FRAAIJE, *LINCS: A linear constraint solver for molecular simulations*, J. Comput. Chem., 18 (1997), pp. 1463–1472.
- [11] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, 8th print of 1st paperback ed., 2007.
- [12] T. J. R. HUGHES, *A note on the stability of newmark’s algorithm in nonlinear structural dynamics*, Int. J. Numer. Meth. Eng., 11 (1977), pp. 383–386.
- [13] A. JAIN, N. VAIDEHI, AND G. RODRIGUEZ, *A fast recursive algorithm for molecular dynamics simulation*, J. Comput. Phys, 106 (1993), pp. 258–268.
- [14] W. L. JORGENSEN, J. CHANDRASEKHAR, J. D. MADURA, R. W. IMPEY, AND M. L. KLEIN, *Comparison of simple potential functions for simulating liquid water*, J. Chem. Phys., 79 (1983), p. 926.
- [15] C. KANE, J. E. MARSDEN, M. ORTIZ, AND M. WEST, *Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems*, Int. J. Numer. Meth. Eng., 49 (2000), pp. 1295–1325.
- [16] J. KEVORKIAN AND J. D. COLE, *Multiple scale and singular perturbation methods*, vol. 114 of Applied Mathematical Sciences, Springer-Verlag, New York, 1996.
- [17] V. KRÄUTLER, W. F. VAN GUNSTEREN, AND P. H. HENENBERGER, *A fast SHAKE algorithm to solve distance constraint equations for small molecules in molecular dynamics simulations*, J. Comput. Chem., 22 (2001), pp. 501–508.
- [18] D. KUHL AND M. A. CRISFIELD, *Energy-conserving and decaying algorithms in non-linear structural dynamics*, Int. J. Numer. Meth. Engng., 45 (1999), pp. 569–599.
- [19] D. KUHL AND E. RAMM, *Constraint energy momentum algorithm and its application to non-linear dynamics of shells*, Computer Methods in Applied Mechanics and Engineering, 136 (1996), pp. 293 – 315.
- [20] P. LAX AND R. RICHTMYER, *Survey of the stability of linear finite difference equations*, Comm. Pure Appl. Math., 9 (1956), pp. 267–293.
- [21] T. LI, A. ABDULLE, AND W. E, *Effectiveness of implicit methods for stiff stochastic differential equations*, Commun. Comput. Phys., 3 (2008), pp. 295–307.

- [22] J. E. MARSDEN AND M. WEST, *Discrete mechanics and variational integrators*, Acta Numerica, (2001), pp. 357–514.
- [23] S. MIYAMOTO AND P. A. KOLLMAN, *Settle: An analytical version of the SHAKE and RATTLE algorithm for rigid water models*, J. Comput. Chem., 13 (1992), pp. 952–962.
- [24] N. M. NEWMARK, *A method of computation for structural dynamics*, Proc. ASCE, 85 (1959), pp. 67–94.
- [25] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [26] J. C. PLATT AND A. H. BARR, *Constraints methods for flexible models*, SIGGRAPH Comput. Graph., 22 (1988), pp. 279–288.
- [27] J.-P. RYCKAERT, G. CICCOTTI, AND H. J. C. BERENDSEN, *Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes*, J. Comput. Phys, 23 (1977), pp. 327–341.
- [28] T. SCHLICK, *Molecular Modeling and Simulation*, Springer, 2nd ed., 2010.
- [29] R. D. SKEEL AND K. SRINIVAS, *Nonlinear stability analysis of area-preserving integrators*, SIAM J. Numer. Anal., 38 (2000), pp. 129–148.
- [30] R. D. SKEEL, G. ZHANG, AND T. SCHLICK, *A family of symplectic integrators: Stability, accuracy, and molecular dynamics applications*, SIAM J. Sci. Comput., 18 (1997), pp. 203–222.
- [31] M. TAO, H. OWHADI, AND J. E. MARSDEN, *Nonintrusive and structure preserving multiscale integration of stiff ODEs, SDEs and Hamiltonian systems with hidden slow dynamics via flow averaging*, Multiscale Model. Simul., 8 (2010), pp. 1269–1324.
- [32] ———, *From efficient symplectic exponentiation of matrices to symplectic integration of high-dimensional Hamiltonian systems with slowly varying quadratic stiff potentials*. arXiv:1006.4659. Accepted by Appl. Math. Res. Express, 2011.
- [33] D. TERZOPOULOS, J. PLATT, A. BARR, AND K. FLEISCHER, *Elastically deformable models*, SIGGRAPH Comput. Graph., 21 (1987), pp. 205–214.
- [34] D. VAN DER SPOEL, P. J. VAN MAAREN, AND H. J. C. BERENDSEN, *A systematic study of water models for molecular simulation: Derivation of water models optimized for use with a reaction field*, J. Chem. Phys., 108 (1998), p. 10220.
- [35] J. M. WENDLANDT AND J. E. MARSDEN, *Mechanical integrators derived from a discrete variational principle*, Phys. D, 106 (1997), pp. 223–246.

- [36] A. WITKIN, K. FLEISCHER, AND A. BARR, *Energy constraints on parameterized models*, SIGGRAPH Comput. Graph., 21 (1987), pp. 225–232.
- [37] W. L. WOOD AND M. E. ODUOR, *Stability properties of some algorithms for the solution of nonlinear dynamic vibration equations*, Commun. Appl. Numer. Methods, 4 (1988), pp. 205–212.